ON THE INCOMPLETE CHOLESKY DECOMPOSITION OF A CLASS OF PERTURBED MATRICES

GÉRARD MEURANT*

Abstract. We consider how to cheaply compute an incomplete Cholesky decomposition of symmetric perturbed matrices $C = \epsilon I + A$ with a small ϵ when knowing an incomplete decomposition of A. Numerical examples are provided that show the effectiveness of the proposed approach.

Key words. Incomplete factorization, conjugate gradient.

AMS subject classification. 65F50

1. Introduction. In this paper we are mainly concerned with the incomplete Cholesky decomposition of symmetric M–matrices

$$C = \epsilon I + A$$

where A arises from the discretization of an elliptic partial differential equation, ϵ being a "small" positive real parameter. Such problem arise, for instance, from discretizing parabolic equations.

As a model problem we can use the two–dimensional heat equation,

$$\frac{\partial u}{\partial t} - \Delta u = f,$$

in the unit square with Dirichlet boundary conditions and an initial condition $u(x, 0) = u_0(t)$. We discretize in space with finite differences with a stepsize h and a time implicit scheme. Then, we obtain

$$(\frac{I}{k} + \frac{1}{h^2}A)u^{n+1} = \frac{u^n}{k} + f^{n+1},$$

where k is the time step and $1/h^2$ A is the matrix of the corresponding elliptic problem. For some problems it makes sense to choose $k \simeq h$. After multiplication by h^2 the matrix of the problem is $C = kI + A$ where k is "small".

The matrix C being symmetric positive definite, we would like to solve the linear system at each time step with the preconditioned conjugate gradient algorithm. A very popular preconditioner is the incomplete Cholesky decomposition without any fill–in IC(1,1) (sometimes also denoted as IC(0)), c.f. [3], [4] or [5], [6] for a review. Usually the time step k is small to obtain the convergence of the approximation. Therefore, it is interesting to know if one can compute an approximation of the incomplete decomposition of C when knowing the decomposition of A. Moreover, very often the time step is not constant, therefore one cannot compute the decomposition of C once for all. It has to be recomputed at each time step. Hence, it would be interesting to find a way to cheaply update the incomplete decomposition from one time step to the next.

Matrices of this type have been considered in previous works, mainly to stabilize the incomplete factorization of A, see [2], [1], when the straightforward factorization of A gives some small pivots.

*CEA/DIF, DCSA/ED, BP 12, 91680 Bruyères le Chatel, France. (meurant@bruyeres.cea.fr)

In section 2, we describe algorithms corresponding to perturbations of order 0 and 1 in ϵ. Section 3 give numerical examples for several problems with comparisons between the incomplete decomposition of C, those of A and the SSOR preconditioning. For a definition of these algorithms, see [5]. We will also give some results for problems not arising from parabolic PDEs, even for cases where A is not an M–matrix. Then in section 4, we apply the previous results to the solution of the heat equation. Section 5 deals with another problem where $C = I + \epsilon A$.

2. Approximate preconditioners. We first recall the first step of the incomplete Cholesky decomposition of a matrix C whose elements are denoted $c_{i,j}$. Let G be a set of indices corresponding, for instance, to the non–zero structure of A and

$$C = C_1 = \begin{pmatrix} c_{1,1} & c_1^T \\ c_1 & E_2 \end{pmatrix} = \begin{pmatrix} c_{1,1} & f_1^T \\ f_1 & E_2 \end{pmatrix} - \begin{pmatrix} 0 & r_1^T \\ r_1 & 0 \end{pmatrix} = M_1 - R_1,$$

with

$$c_1 = f_1 - r_1,$$

$$(f_1)_i = 0, \text{ if } (i,1) \notin G \Rightarrow (r_1)_i = -(c_1)_i,$$

$$(f_1)_i = (c_1)_i, \text{ if } (i,1) \in G \Rightarrow (r_1)_i = 0.$$

Then, we factorise M_1

$$M_1 = \begin{pmatrix} c_{1,1} & 0 \\ l_1 & I \end{pmatrix} \begin{pmatrix} c_{1,1}^{-1} & 0 \\ 0 & C_2 \end{pmatrix} \begin{pmatrix} c_{1,1} & l_1^T \\ 0 & I \end{pmatrix} = L_1 \Sigma_1 L_1^T.$$

We obtain

$$\begin{aligned} l_1 &= f_1, \\ C_2 &= E_2 - \frac{1}{c_{1,1}} f_1 f_1^T. \end{aligned}$$

The next step is applying the same decomposition on C_2,

$$C_2 = \begin{pmatrix} c_{2,2}^{(2)} & c_2^T \\ c_2 & E_3 \end{pmatrix} = \begin{pmatrix} c_{2,2}^{(2)} & f_2^T \\ f_2 & E_3 \end{pmatrix} - \begin{pmatrix} 0 & r_2^T \\ r_2 & 0 \end{pmatrix} = M_2 - R_2,$$

where f_2 is obtained from c_2 by setting to zero the elements for which the indices $(i,2)$ do not belong to G. Let

$$L_2 = \begin{pmatrix} c_{1,1} & 0 \\ 0 & \begin{pmatrix} c_{2,2}^{(2)} & 0 \\ l_2 & I \end{pmatrix} \end{pmatrix}.$$

Note that $l_2 = f_2$ and that we shall never throw away a diagonal entry.

At the end of the second step, we have

$$C = L_1 L_2 \Sigma_2 L_2^T L_1^T - L_1 \begin{pmatrix} 0 & 0 \\ 0 & R_2 \end{pmatrix} L_1^T - R_1,$$

but

$$L_1 L_2 = \begin{pmatrix} c_{1,1} & 0 \\ l_1 & \begin{pmatrix} c_{2,2}^{(2)} & 0 \\ l_2 & I \end{pmatrix} \end{pmatrix} \quad \text{and } L_1 \begin{pmatrix} 0 & 0 \\ 0 & R_2 \end{pmatrix} L_1^T = \begin{pmatrix} 0 & 0 \\ 0 & R_2 \end{pmatrix}.$$

Therefore, we have a factor with the desired structure and we can go on as long as the pivots $c_{i,i}^{(i)}$ are non zero. It has been proven that IC is feasible whatever the set G is when C is an H–matrix. We apply this algorithm to $C = \epsilon D + A$ where D is a diagonal matrix with non zero diagonal elements d_i. We use a matrix $D \neq I$ because with the order 1 algorithm to be described soon, the diagonal entries are going to be modified so that we do not get the identity for all steps of the algorithm. We have

$$c_{1,1} = \epsilon d_1 + a_{1,1},$$

$$l_1 = f_1,$$

$$C_2 = \epsilon D_2 + A_2 - \frac{1}{\epsilon d_1 + a_{1,1}} f_1 f_1^T,$$

where these matrices are defined by

$$C = \begin{pmatrix} \epsilon d_1 + a_{1,1} & a_1^T \\ a_1 & \epsilon D_2 + A_2 \end{pmatrix}.$$

Rather than computing C_2 exactly, we would like to use an asymptotic expansion related to ϵ for the ratio. For the order "0" we have

$$C_2 = \epsilon D_2 + A_2 - \frac{1}{a_{1,1}} f_1 f_1^T,$$

that is to say we add ϵD_2 to what we would have obtained for the incomplete decomposition of A.

The order "1" gives

$$C_2 = \epsilon \left(D_2 + \frac{d_1 f_1 f_1^T}{a_{1,1}^2} \right) + A_2 - \frac{1}{a_{1,1}} f_1 f_1^T.$$

To motivate the choices we are going to do later on, let us look at what we get for a 2D finite difference matrix. As an example, we take the Poisson equation in a square with a mesh size $h = 1/(m+1)$. This leads to a matrix of order $n = m^2$ which can be written blockwise as

$$A = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix},$$

with blocks of order m

$$T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

The matrix A has five non–zero diagonals. For a generic row i and more general problems, the non–zero coefficients are $a_{i,i-m}, a_{i,i-1}, a_{i,i}, a_{i+1,i}, a_{i+m,i}$. When we compute the incomplete decomposition IC(1,1) (with no fill–in) of this matrix obtaining $M = (\Sigma + L)\Sigma^{-1}(\Sigma + L^T)$, it is easy to see that we only need to compute the diagonal matrix Σ whose elements are denoted $\sigma_{i,i}$. The matrix L is the strictly lower triangular part of A. The diagonal elements are given by

$$\sigma_{i,i} = a_{i,i} - \frac{a_{i,i-1}^2}{\sigma_{i-1,i-1}} - \frac{a_{i,i-m}^2}{\sigma_{i-m,i-m}}.$$

In this formula, entries of A (resp. Σ) whose indices do not exist are taken to be 0 (resp. 1). The two ratios arises from the two non–zero elements in each column of L and in each row of L^T the upper triangular part of A.

Let us look at what we obtain in the first step of the decomposition of $C = \epsilon D + A$. For order "0" there is no problem as we have just to compute $\sigma_{i,i}$ as before and to add ϵd_i.

Handling order "1" is a little more tricky. We obtained

$$C_2 = \epsilon \left(D_2 + \frac{d_1 f_1 f_1^T}{a_{1,1}^2} \right) + A_2 - \frac{1}{a_{1,1}} f_1 f_1^T.$$

To what we would have obtained for A, that is

$$A_2 - \frac{1}{a_{1,1}} f_1 f_1^T$$

which corresponds to the computation of $\sigma_{i,i}$, we have to add the correction of order ϵ. That is

$$\epsilon \left(D_2 + \frac{d_1 f_1 f_1^T}{a_{1,1}^2} \right).$$

Since there are only two non–zero elements in f_1 for the model problem, the outer product $f_1 f_1^T$ gives two diagonal modifications for indices $(2,2)$ et $(m+1, m+1)$ (when these terms exist). Therefore, we add

$$\epsilon \left(d_2 + d_1 \frac{a_{2,1}^2}{a_{1,1}^2} \right)$$

to the element of index $(2,2)$ and

$$\epsilon \left(d_{m+1} + d_1 \frac{a_{m+1,1}^2}{a_{1,1}^2} \right)$$

to the element of index $(m+1, m+1)$. This modifies the diagonal terms of order ϵ. Therefore we see that the modification of the diagonal terms is recursive. This means that there is no gain from doing the decomposition of C from scratch. Consequently, we decided to bypass the recursion. We only apply the modifications to the initial D. For example the correction of the element $(3,3)$ will be

$$\epsilon \left(d_3 + d_2 \frac{a_{3,2}^2}{\sigma_{2,2}^2} \right).$$

This will work when ϵ is small but not with larger values of the parameter. In the latter case we have

$$\frac{1}{\epsilon d_1 + a_{1,1}} \simeq \frac{1}{\epsilon d_1}.$$

Therefore, in the formulas for order "1" we replace $\sigma_{i,i}$ by $\sigma_{i,i} + \epsilon d_i$. This does not make too much difference when ϵ is small and gives asymptotically the exact answer when it is large. For instance, the final correction for the element $(3,3)$ in the second step will be

$$\epsilon \left(d_3 + d_2 \frac{a_{3,2}^2}{(\sigma_{2,2} + \epsilon d_2)^2} \right).$$

This algorithm is what we call order "1" in the numerical experiments. This has the additional advantage that all the corrections can proceed in parallel. In this way we get rid of the recursion of the incomplete Cholesky factorization which is not easily parallelizable. However, note that there is still a recursion when solving the triangular systems.

In the case of a matrix arising from a diffusion equation with constant coefficient we can perform an asymptotic analysis of the diagonal elements of the incomplete factors. If $a_{i,i} = a, a_{i,i-1} = b, a_{i,i-m} = c$, then the elements of the incomplete decomposition $\sigma_{i,i}$ (within a block) converges rapidly to σ

$$\sigma = \frac{a+s}{2}, \quad s = \sqrt{a^2 - 4(b^2 + c^2)}.$$

If the diagonal coefficients of A are perturbed by ϵ we obtain a limit $\bar{\sigma}$

$$\bar{\sigma} \simeq \frac{a+s}{2} + \frac{\epsilon}{s} \frac{a+s}{2}.$$

The fact that the difference between the exact factorization and the approximate one is small is also illustrated in figure 2.1 where we show the relative differences between the exact and approximate order "0" values of the second block of diagonal coefficients for the perturbed Poisson equation with a 30×30 mesh as a function of the relative index in the block. The solid line is $\epsilon = 10^{-2}$, the dashed line $\epsilon = 10^{-1}$, the dot–dashed line $\epsilon = 1$, the dotted line $\epsilon = 10$, the plus signs $\epsilon = 100$ and the circles $\epsilon = 1000$. Over all values of ϵ the maximum relative difference is 4%.

This technique can be generalized to any non–zero structure for A. For order "0" we just add ϵD to the diagonal. For the order "1" we modify only the diagonal, neglecting the recursiveness and we add an order of ϵ term in the denominator to obtain the correct behaviour when ϵ is large. The method can also be applied to matrices arising from finite element methods. In case mass lumping is used we can do exactly the same thing. If the mass matrix is not diagonal, we can do modifications to the non zero entries but neglecting the recursiveness that is using the initial values of the entries of the mass matrix.

3. Numerical experiments. We denote by IC the incomplete Cholesky decomposition without fill–in. We use several examples to compare the following preconditioners:

1. IC for $C = \epsilon I + A$
2. approximate IC of C order "0"

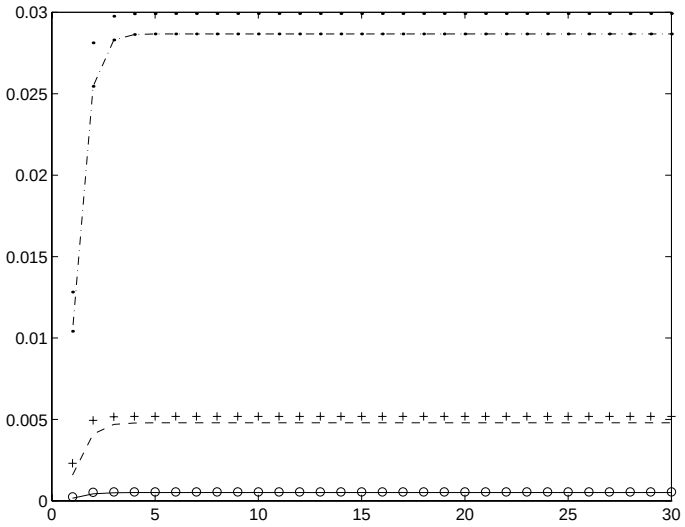

FIG. 2.1. *Relative differences between diagonal coefficients*

TABLE 1
Poisson problem

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
320	2	3	3	40	2
80	3	4	4	39	3
20	4	5	5	35	4
5	6	6	6	26	7
1.25	10	10	10	14	11
0.325	17	17	17	15	19
$7.812\ 10^{-2}$	26	26	26	25	29
$1.953\ 10^{-2}$	32	32	32	32	39
$4.882\ 10^{-3}$	34	34	34	34	40

3. approximate IC of C order "1"
4. IC for A
5. SSOR with $\omega = 1$.

We first consider matrices arising from diffusion equations in the unit square with homogeneous Dirichlet boundary conditions. The first example is the Poisson equation we described in section 2. We solve a linear system whose solution is $x = \{1, 1, \ldots, \}^T$. The initial iterate is chosen at random and we stop the iterations as soon as the relative norm residual is less 10^{-10}. We use a regular 30×30 cartesian mesh. We start with $\epsilon = 320$ and we divide it by 4 several times. The results are given in table 1.

For this example, the two approximate decompositions give almost the same number of iterations as the "exact" incomplete decomposition. The results of the approximate decompositions are also quite good for large ϵ's. This is linked to the fact that in this case the matrix is diagonally dominant and that we add a proper correction for the order "1" . We also note that in this case the results of the order "1" are the same as those of order "0". Of course, when ϵ is small we obtain the same number of iterations when we only use IC(A).

The second example is a diffusion problem with discontinuous diffusion coefficients and Dirichlet boundary conditions on the unit square. The diffusion coefficient is 1000

TABLE 2
Diffusion problem with discontinuous coefficients

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
320	16	16	16	148	19
80	24	24	24	171	28
20	29	29	29	129	35
5	32	32	32	79	37
1.25	33	33	33	44	38
0.325	34	34	34	36	40
$7.812\ 10^{-2}$	37	37	37	37	43
$1.953\ 10^{-2}$	39	39	39	39	45
$4.882\ 10^{-3}$	41	41	41	41	47

TABLE 3
Diffusion problem with anisotropic coefficients

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
320	4	8	8	143	6
80	5	12	9	92	10
20	7	15	11	53	18
5	10	14	12	28	33
1.25	15	15	15	16	57
0.325	25	26	26	25	97
$7.812\ 10^{-2}$	34	34	34	33	126
$1.953\ 10^{-2}$	38	38	38	38	134
$4.882\ 10^{-3}$	39	39	39	39	136

in $[1/4, 3/4] \times [1/4, 3/4]$ and 1 elsewhere. We use the same mesh and parameters as before. Results are provided in table 2. The conclusions are the same as for the first example.

The third example is a diffusion problem with an anisotropic coefficient and Dirichlet boundary conditions on the unit square. The x diffusion coefficient is 100 in $[1/4, 3/4] \times [0, 1]$ and 1 elsewhere. The y diffusion coefficient is 1 everywhere. The parameters are the same as before. Results are given in table 3.

This is a difficult problem and we can see there are differences between IC and the approximate decompositions. For this problem we get an improvement when going from order "0" to order "1" for middle range values of ϵ . However, the differences are small and we can still conclude that it could be useful to use the approximate decomposition.

We now consider some matrices from the Harwell–Boeing collection or from the Boeing collection arising from the Tim Davis' Web site (`http://www.cise.ufl.edu`). We had to normalize some of these matrices in order for the perturbations ϵI to be meaningful. We use the following examples:

1. 1138-bus. An admittance matrix of order 1138 with 4054 non–zeros. It was normalized. Note that the unperturbed matrix is close to being singular.

2. bcsstk01. A stiffness matrix of order 48 with 400 non-zeros. It was normalized. This matrix is not diagonally dominant, nor an M–matrix, but nevertheless positive definite.

3. gr3030. A matrix arising from a nine point approximation to the Laplacian on the unit square with a 30×30 mesh. It has order 900 and 7744 non–zeros.

4. msc00726. A matrix of order 726 with 34518 non-zeros from Nastran. This matrix was normalized.

TABLE 4
1138-bus

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
1000	1	2	2	205	1
250	2	3	3	249	2
62.5	2	3	3	249	2
15.625	2	4	4	242	3
3.51	3	5	5	215	4
0.98	6	8	8	163	6
0.24	10	14	13	110	10
0.0610	18	22	21	93	19
0.0015	30	34	33	86	34
0.0038	46	48	48	81	62
$9.54\ 10^{-4}$	65	65	65	80	108
$2.38\ 10^{-4}$	83	83	83	86	175
$5.963\ 10^{-5}$	101	101	101	102	250
$1.490\ 10^{-5}$	114	114	114	114	320

TABLE 5
bcsstk01

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
1000	1	2	2	31	1
250	2	3	3	30	2
62.5	2	3	3	31	2
15.625	2	4	4	29	2
3.51	3	5	5	26	4
0.98	4	7	7	19	6
0.24	6	8	8	13	9
0.0610	7	10	10	12	14
0.0015	10	11	11	12	18
0.0038	12	12	12	13	23
$9.54\ 10^{-4}$	13	13	13	14	24
$2.38\ 10^{-4}$	14	14	14	14	24
$5.963\ 10^{-5}$	14	14	14	14	24
$1.490\ 10^{-5}$	14	14	14	14	24

We ran the different problems and preconditioners with values of ϵ ranging from 1000 to $1.49\ 10^{-5}$. In tables 4 to 7 we report the number of iterations obtained by using a stopping criterion of 10^{-6} on the relative residual norm in CG.

For 1138-bus there is no gain by using order "1" over order "0". The SSOR preconditioner gives better results than the approximate ICs for large ϵ but, for small ones, the SSOR results are much worst. We can see that although this is a different kind of problem the approximate preconditioners still give very good results. Note that ϵ has to be very small for having good results with IC(A). The conclusions for bcsstk01 are the same as for the previous example. For the last two problems gr3030 and msc00726 we can also draw the same conclusions. We can obtain a very good preconditioner by just modifying the diagonal elements in a parallel way. A general remark is that using IC(A) is fine when $\epsilon \rightarrow 0$ but with the perturbed factorizations we can also get good results when ϵ is large. Moreover, the results are generally better than using straight SSOR.

4. Application to the heat equation. We consider the following problem:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f, \quad \text{in } \Omega =]0,1[^2$$

TABLE 6
gr3030

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
1000	2	2	2	25	2
250	2	3	3	24	2
62.5	2	3	3	23	2
15.625	3	4	4	19	4
3.51	5	5	5	12	6
0.98	7	7	7	7	10
0.24	11	11	11	14	15
0.0610	14	14	14	16	19
0.0015	16	16	16	17	22
0.0038	17	17	17	17	23
$9.54\ 10^{-4}$	17	17	17	17	23
$2.38\ 10^{-4}$	17	17	17	17	23
$5.963\ 10^{-5}$	17	17	17	17	23
$1.490\ 10^{-5}$	17	17	17	17	23

TABLE 7
msc00726

ϵ	IC(C)	order "0"	order "1"	IC(A)	SSOR
1000	1	2	2	44	1
250	2	3	3	44	2
62.5	2	3	3	43	2
15.625	2	4	4	41	3
3.51	3	6	6	34	4
0.98	5	8	8	23	6
0.24	8	9	9	13	10
0.0610	12	12	12	12	17
0.0015	18	18	18	18	25
0.0038	26	26	26	26	38
$9.54\ 10^{-4}$	30	30	30	30	40
$2.38\ 10^{-4}$	31	31	31	31	41
$5.963\ 10^{-5}$	31	31	31	31	41
$1.490\ 10^{-5}$	31	31	31	31	41

with Dirichlet boundary conditions and a given initial condition.

We discretize with finite differences and a time implicit scheme as shown in the introduction. We use the conjugate gradient algorithm to solve the linear system we obtain at each time step.

We solved a problem whose exact solution is $u = (1 + t^3)\sin(\pi x)\sin(\pi y)$. We use 30 discretization points in each direction and $k = h$. We obtained exactly the same total number of CG iterations using either the "exact" incomplete Cholesky decomposition recomputed at each time step or the approximate one of order "0". Therefore this shows that in this case it is useless to recompute the decomposition at each time step, we just have to update the diagonal of the decomposition of A which is computed during the initialization phase. Of course, updating the incomplete factorization was much cheaper than recomputing at every time step. We do not give any computer times since this computation was done using Matlab for which computer times depend very much on how the programs are written.

5. Perturbation of a diagonal matrix. For completeness, we now consider the incomplete Cholesky decomposition of a matrix $C = I + \epsilon A$ although this case has much less practical applications than the case we handle in section 2. Let $c_1 = f_1 - r_1$,

TABLE 8
Diffusion problem with anisotropic coefficients

ϵ	IC(C)	order "1"	order "2"	ICd	SSOR
320	40	138	86	40	138
80	39	135	85	39	135
20	35	129	82	36	129
5	29	111	71	29	111
1.25	18	70	43	18	70
0.325	11	40	25	14	40
$7.812\ 10^{-2}$	7	22	13	15	22
$1.953\ 10^{-2}$	5	12	7	13	12
$4.882\ 10^{-3}$	4	7	4	9	7

$l_1 = \epsilon f_1$. For the first step we get

$$C_2 = I + \epsilon A_2 - \frac{1}{1+\epsilon a_{1,1}} l_1 l_1^T = I + \epsilon (A_2 - \frac{\epsilon}{1+\epsilon a_{1,1}} f_1 f_1^T).$$

Now we use an asymptotic expansion of the ratio. The order 1 expansion gives $l_1 = \epsilon f_1$, $C_2 = I + \epsilon A_2$. This is nothing else than the SSOR preconditioner with $\omega = 1$. For the order 2 expansion we obtain

$$C_2 = I + \epsilon A_2 - \epsilon^2 f_1 f_1^T.$$

Looking at this formula we may already think that this correction cannot give good results for large ϵ. It may even happen that the preconditioner is not positive definite. To obtain something which can work for both small and large ϵ it makes sense to use a weighting factor. Therefore, we propose to use

$$C_2 = I + \epsilon A_2 - \frac{\epsilon^2}{\epsilon a_{1,1} + 1} f_1 f_1^T.$$

This is what is denoted as order "2" in the results. As we did in section 2 we neglect the recursion when we compute the diagonal corrections.

Another way is to use the incomplete decomposition of A which computes a diagonal d and to set the diagonal elements of the approximate decomposition to $1 + \epsilon d_{i,i}$. We denote this method as ICd. We consider the third example from section 3 for which results are given in table 8.

The results in table 8 show that the order "1" approximation indeed gives the same results as SSOR with $\omega = 1$. The order "2" gives better results than order "1". For small ϵ it gives almost the same results as IC(C). Finally, ICd always gives good results. There are not too far from those of IC(C) for all values of ϵ although order "2"gives better results for very small ϵ. Therefore, ICd seems to be the method of choice for this case.

6. Conclusion. In this paper we have shown that we can efficiently and easily compute Incomplete Cholesky–like preconditioners of $\epsilon I + A$ and $I + \epsilon A$ when we know the incomplete decomposition of A. The proposed methods work for a large range of values of the perturbation parameter ϵ. This approximate factorizations can be useful when solving time dependent partial differential equations since we do not have anymore to recompute the factorization at each time step but only to update it which is a cheap and parallel operation.

Acknowledgments. Thanks to Henk van der Vorst for helpful comments on a preliminary draft of this paper.

REFERENCES

[1] H.C. Elman, *A stability analysis of incomplete LU factorizations*, Math. Comp., vol 47, no 175, (1986), pp 191-217.
[2] T.A. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Math. Comp., vol 34, no 150, (1980), pp 473-497.
[3] J.A. Meijerink and H. Van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M–matrix*, Math. Comp., vol. 31, no 137, (1977), pp 148–162.
[4] J.A. Meijerink and H. Van der Vorst, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*, J. Comp. Physics, vol. 44, (1981), pp 134–155.
[5] G. Meurant, *Computer solution of large linear systems*, North–Holland, (1999).
[6] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Corp., (1996).